# Project Report

Odd Semester (2021)

---


BINUS UNIVERSITY

BINUS INTERNATIONAL

**Assignment Cover Letter**

**(Individual Work)**

**Student Information:**

**Surname:** Koesmanto **Given Name:** Edward Alvin **Student ID Number:** 2501963141

**Course Code** : COMP6699001 **Course Name** : Object Oriented Programming

**Class** : L2BC **Lecturer** : Jude Joseph Lamug Martinez, MCS

**Type of Assignments:** Term Final Project

**Submission Pattern**

**Due Date** :10 June 2022 **Submission Date** : 10 June 2022

The assignment should meet the below requirements.

1. Assignment (hard copy) is required to be submitted on clean paper, and (soft copy) as per lecturer's instructions.
2. Soft copy assignment also requires the signed (hardcopy) submission of this form, which automatically validates the softcopy submission.
3. The above information is complete and legible.
4. Compiled pages are firmly stapled.
5. Assignment has been copied (soft copy and hard copy) for each student ahead of the submission.

**Plagiarism/Cheating**

BiNus International seriously regards all forms of plagiarism, cheating, and collusion as academic offenses which may result in severe penalties, including loss/drop of marks, course/class discontinuity, and other possible penalties executed by the university. Please refer to the related course syllabus for further information.

**Declaration of Originality**

By signing this assignment, I understand, accept, and consent to BiNus International terms and policy on plagiarism. Herewith I declare that the work contained in this assignment is my own work and has not been submitted for the use of assessment in another course or class, except where this has been notified and accepted in advance.

Signature of Student: Edward Alvin Koesmanto

# Project Specification

## Overview

This program is a revamped Hangman game. The goal of the game is to guess the word correctly before the number of tries is finished. In this game, the user is allowed to make 9 tries including the character guessed correctly hence the name revamped.

1. Program Input:
   a. A mouse click that signifies the letter chosen.
   b. A mouse click that signifies the exit of the program
2. Program Output:
   a. A dropdown menu consisting of the letters that hasn't been guessed yet.
   b. A sequence of letters that is correct according to the word.
   c. A game over window once the tries has been used up.

## Libraries used

1. Swing
   - Used to make the GUI of the game.
2. Scanner
   - Used to take input from the user.
3. File
   - Used to retrieve elements from an external file.

# Solution Design

Files involved are WordsBank.java, ChosenWord.java, Game.java and Main.java

## WordsBank.java

This java file is an interface which consist of the function to get a random word from a txt file.

```
public interface WordsBank {
    You, 4 days ago | 1 author (You)
    // // function to get a random word
    public String getRandomWord();
}
```

# ChosenWord.java

This java file has the main purpose to check the word taken from the txt file.

```
public class ChosenWord {
    private String word;
    private boolean[] charsGuessed;
```

Properties of this class is a string (word) and boolean array (charsGuessed).

```
//// Constructor to convert the word into all lowercase letters
public ChosenWord(String word) {
    this.word = word.toLowerCase();
    //! create new array of boolean to check if the character inputted is correct as the character
    //! in the position of the answer
    charsGuessed = new boolean[word.length()];
}
```

The constructor of this class converts the word from the txt file into all lowercase and creates an array with the length of the word as size.

```
//// Function to check if the word is already guessed or not
public boolean isEntireWordGuessed() {
    //! if there is false in the answer, return false
    for (boolean b : charsGuessed) {
        if (!b) {
            return false;
        }
    }
    return true;
}
```

The isEntireWordGuessed() function's main purpose is to check if all the letters of the word has been guessed or not. It will iterate through the array and return false if one of the letter has not been guessed.

```java
//// Function to alocate index of the character guessed
public void charGuess(char guess) {
    int index = word.indexOf(guess);

    while (index >= 0) {
        charsGuessed[index] = true;
        index = word.indexOf(guess, index + 1);
    }
}
```

The charGuess() function is used to get the index of the characters that have been guessed.

```java
@Override
//// Function to convert char into string
public String toString() {
    StringBuilder formattedWord = new StringBuilder();

    for (int i = 0; i < word.length(); i++) {
        if (charsGuessed[i]) {
            formattedWord.append(word.charAt(i));
        } else {
            formattedWord.append(c: '_');
        }

        formattedWord.append(c: ' ');
    }
    return formattedWord.toString();
}
```

This function is to convert characters into strings to be displayed.

## Game.java

This java file stores most of the logic of the game.

```java
public class Game extends ChosenWord implements WordsBank{
    You, 3 seconds ago | 1 author (You)
    public Game(String word) {
        super(word);
    }

    private int numberOfGuesses;
    private int tries;
    private String[] words;
    private String unguessedCharacters;
    private ChosenWord chosenWord;
    private JFrame frame = new JFrame(title: "Revamped Hangman");
    private JFrame frame1 = new JFrame(title: "Game Over");
```

This java file implements WordsBank and inherit from ChosenWord to get a random word from the txt file and the properties are as shown.

```java
@Override
//// Function to get a random word from "words.txt";
public String getRandomWord() {
    words = new String[5757];
    //// Source from: https://github.com/SimonVutov/WordleGUI/blob/main/GUI.java
    try {
        File myObj = new File(pathname: "words.txt");
        Scanner myReader = new Scanner(myObj);
        int counter = 0;
        while (myReader.hasNextLine()) {
            String data = myReader.nextLine();
            //! add data into array
            words[counter] = data;
            counter++;
        }
        myReader.close();
    } catch (FileNotFoundException f) {
        System.out.println(x: "An error has occured");
        f.printStackTrace();
    }
    // // returns a random word from the txt file
    return words[(int)(Math.random() * (words.length - 1))];
}
```

The getRandomWord() function taken from WordsBank.java is implemented here. This function reads the whole words.txt

```java
//// Function to start the game
public void startNewGame() {
    this.unguessedCharacters = "abcdefghijklmnopqrstuvwxyz";

    numberOfGuesses = 0;
    tries = 8;
    this.chosenWord = new ChosenWord(getRandomWord());

    inputUserLetterGuess();
}
```

This function initiates a new game and prompts the user to input a letter to guess

```java
//// Function to validate the user's guesses
private void inputUserLetterGuess() {
    //! create an array of the unguessed characters
    Character[] charactersArray = new Character[unguessedCharacters.length()];

    for(int i = 0; i < charactersArray.length; i++) {
        charactersArray[i] = Character.valueOf(unguessedCharacters.charAt(i));
    }

    //! display a window of the unguessed characters
    Character guessedLetter = (Character) JOptionPane.showInputDialog(frame, message: "What letter do you want to guess?",
                                                                      title: "Letter Guess",
                                                                      JOptionPane.QUESTION_MESSAGE,
                                                                      icon: null,
                                                                      charactersArray,
                                                                      //! sets the first unguessed letter of the alphabet as a placeholder
                                                                      charactersArray[0]);

    //! if all letters have been guessed or tries are 0, exit the program
    if (guessedLetter == null || tries == 0) {
        GameOver(frame1);
        return;
    }

    handleUserLetterGuess((guessedLetter));

    displayUserGuessResults(frame);
}
```

This function takes the user input and validate it with the random word.

```java
private void handleUserLetterGuess(char guessedChar) {
    numberOfGuesses++;
    tries--;

    removeOptionalCharGuess(guessedChar);

    chosenWord.charGuess(guessedChar);
}
```

This function validates the user's guess and increment the number of guesses and decrement the number of tries.

```
//// Function to remove guessed characters from the option
private void removeOptionalCharGuess(char guessedChar) {
    //! replace the guessed characters as blank
    unguessedCharacters = unguessedCharacters.replace(Character.toString(guessedChar), replacement: "");
}
```

This function removes the guessed letter.

```
//// Function to show progress of the guesses
private void displayUserGuessResults(JFrame frame){
    //! shows a character if the character guessed is correct
    JLabel wordStartLabel = new JLabel("After your guess: " + chosenWord.toString());
    JLabel triesLeft = new JLabel("Tries left: " + (tries + 1));
    JButton button = new JButton();

    JPanel panel = new JPanel();
    panel.add(wordStartLabel);
    panel.add(triesLeft);
    panel.add(button);

    //! Sets the window
    frame.add(panel);
    frame.setSize(width: 300, height: 300);
    frame.setLocationRelativeTo(c: null);

    //! makes sure the exit is working smoothly
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    //! makes the frame visible
    frame.setVisible(b: true);
```

This function displays the GUI.

```
//// Checks if the word is completely exposed
if (!chosenWord.isEntireWordGuessed()){
    //! If word isn't completely guessed, continue guessing
    button.addActionListener(e -> {
        frame.remove(panel);
        inputUserLetterGuess();
    });

    //! makes a button to continue guessing
    button.setText(text: "Continue guessing");

} else {
    //! If word is completely guessed, show results and give option to start a new game
    JLabel guessesLabel = new JLabel("Congratulations! number of guesses is: " + numberOfGuesses);
    panel.add(guessesLabel);
    button.addActionListener(e -> {
        frame.remove(panel);
        startNewGame();
    });
    button.setText(text: "Start a new game");
}
```

The continuation of displayUserGuessResults.

```java
private void GameOver(JFrame frame) {
    JLabel gameOver = new JLabel(text: "Game Over");

    JPanel panel = new JPanel();

    panel.add(gameOver);
    frame.add(panel);
    frame.setSize(width: 300, height: 200);
    frame.setLocationRelativeTo(c: null);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setVisible(b: true);
    exit();
}
```

This displays the GameOver screen at the end of the game.

```java
//// Function to close the frame on forced exit
private void exit() {
    frame.dispose();
}
```

This function closes the frame.

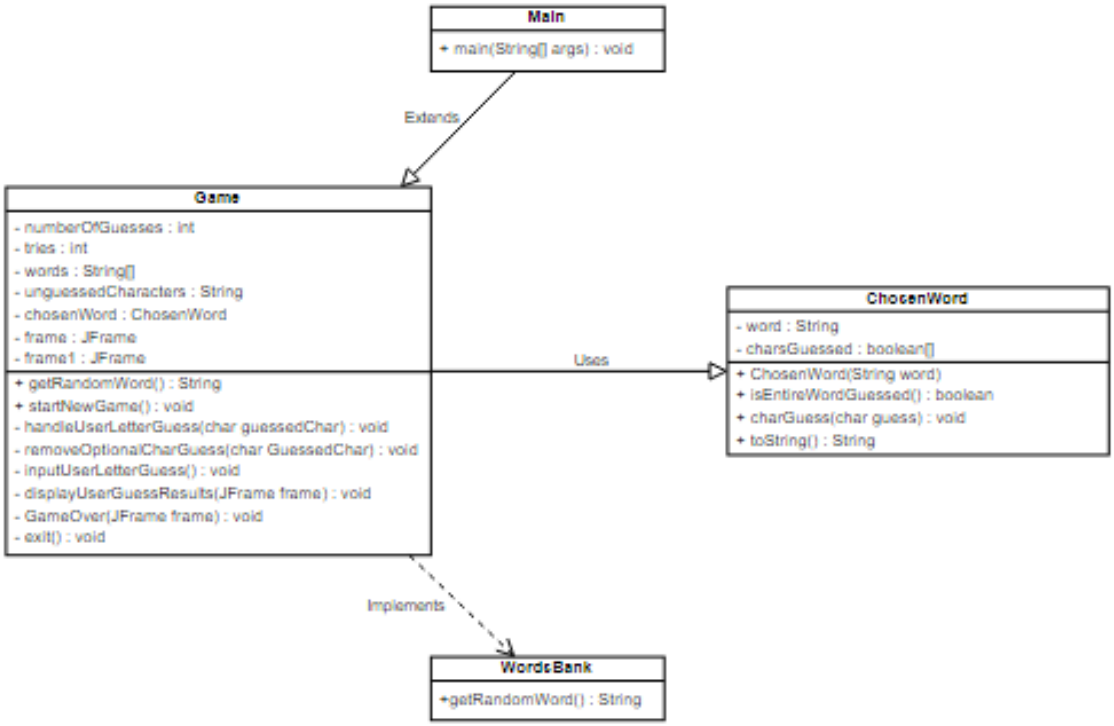
## Main.java

This function runs the program.

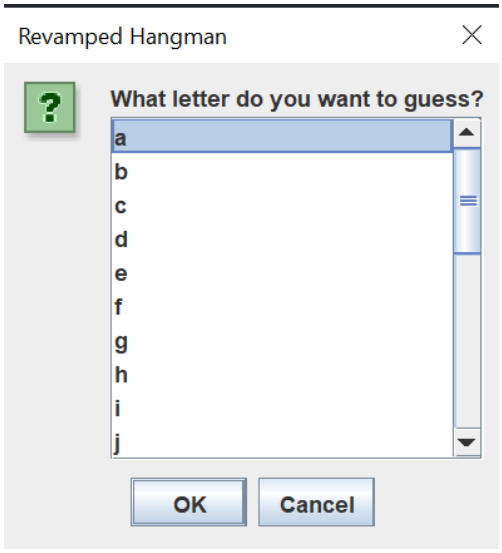
```java
public class Main extends Game{
    Run | Debug | You, 4 days ago | 1 author (You)
    public static void main(String[] args) {
        //! Starts the game
        Game game = new Game();
        game.startNewGame();
    }
}
```

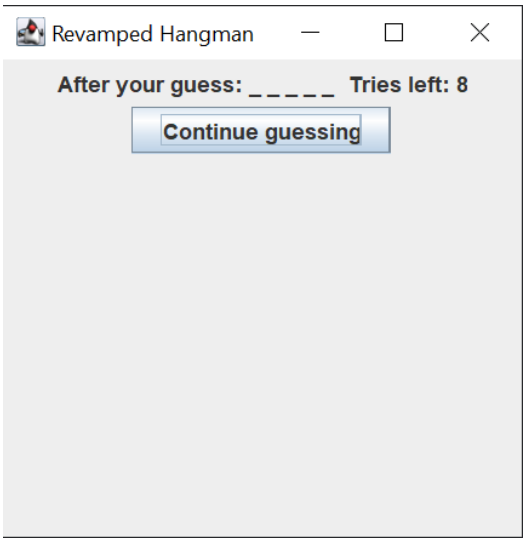
Instantiate the game and starts it.

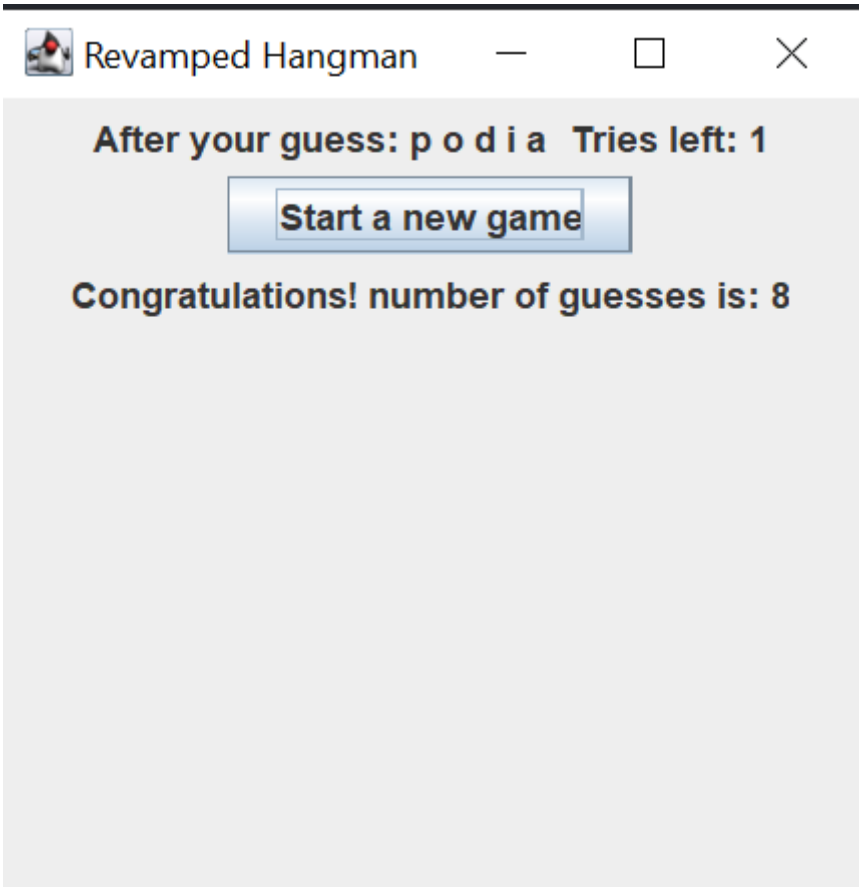
## Class Diagram




## Evidence of working program






## Sources


- https://github.com/SimonVutov/WordleGUI/blob/main/GUI.java
- https://www.geeksforgeeks.org/character-valueof-in-java-with-examples/
- https://www.youtube.com/watch?v=qsGKSB_tR6E
- https://netbeans.apache.org/kb/docs/java/gui-functionality.html
- https://www.guru99.com/java-swing-gui.html